An Enhanced format output design for printf

Document: N 1695

Contents

1 Purpose

The purpose of this document is written in detail zprintf () function design ideas, in order to benefit other programmers use well zprintf () function characteristics to write code series.

2 Glossary, Definitions and Abbreviations

2.1 Glossary and Definitions

Group: in zprintf () function is used to take several shows, if repeated several times to take more, they separated by combined use of this combination separator, the number of repeat inside take number is called Group.

Stem: conversion functions in the source data into a form easily understood when the user data source, depends entirely on that part of a string. Or, every one of a string of conversion remove width compensation, precision compensation, leading symbols that part of a string outside compensation.

2.2 Abbreviations

IMSI: International Mobile Subscriber Identity

3 Overview

Someone may have such doubt, printf () function provided by standard C library is enough, why should we develop an enhanced one?

The early zprintf () function idea is actually very simple, because we frequently need print such structure's content: IP address, IMSI. As we all know, IP address(for V4) is a four bytes integer, usually on PC or on the mainstream of the embedded processor now it is only a machine word. But we can't easily get an intuitive express such as "10.50.1.1" by printf which is provided by standard C library. If we print such a value with integer style, we may get different result on different architecture processor. For example, the value of IP address "10.50.1.1" on X86 PC is 16855562, while on PPC is 171049217. Obviously, it's unimaginable.

With hexadecimal style, it is also difficult to read the value of the IP address at first sight. As on X86 PC, this address value can show 0xa320101, and on PPC is 0x101320a. If a larger number is introduced, we will be crazy. Of course, we may use inet_ntoa() function to translate to a string format, and then display it. But please think, although such requirements are seemly simple, but it increases programmers meaningless workload. We only want to inspect the value, we do not want to assign an extra variable for the inspection, and then take further step to pass the result to the printf() function.

If we merely to display an IP address for such requirements, our printf () function improved analytical desire also not so urgent. But the writer is engaged in the telecommunications industry, this problem in communication system, the development of embedded contradictions are outstanding. Because we are not the IP address waiting to show, we also like IMSI, belonging to some specific location area, etc. Of the value need check structure, When we need to see these structures, we always have to list the members of the structure again and again, and then put into

the printf () function variable and list it wait. This job is a headache. As I will show IMSI address is in memory address and IMSI 8 bytes commonly used, each byte 2 hexadecimal number, and often is low address four is low, the highest IMSI effectively lower the address for next four high effective. As a description of the eight bytes of memory IMSI value as follows:

21 43 65 87 09 21 43 65

So the actual IMSI value is? 1234567890123456 should be. If I use printf () function to display correct such a value, will have to list sixteen, and each parameter to use similar parameters under such expressions:

Imsi[0]&0xf, Imsi[0]>>4,

Every time I want to show that an IMSI value, are written form below:

printf("IMSI: %x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x\n",

Imsi[0]&0xf, Imsi[0]>>4, Imsi[1]&0xf, Imsi[1]>>4, Imsi[2]&0xf, Imsi[2]>>4, Imsi[3]&0xf, Imsi[3]>>4, Imsi[4]&0xf, Imsi[4]>>4, Imsi[5]&0xf, Imsi[5]>>4, Imsi[6]&0xf, Imsi[6]>>4, Imsi[7]&0xf, Imsi[7]>>4);

It is easy to see how the code that no human. Otherwise, they also like IP address as processing, a conversion functions, from the conversion function display the results.

Although a conversion functions, said nothing. But such examples in communication system, because to enumerate in communication system, communication protocol is often by format of analytical. Most using C language writing code, are directly applying agreement, will define the agreement with the corresponding structure code conversion, and there are many bits can use case, every field that this value, the developer shall be similar to this list.

In addition to the above the unfavorable factors, early C standard haven't defined shape as snprintf () function, its system library core conversion is an open interfaces. It's not safe. I use the following definitions:

char buf[10];

sprintf(buf, "hello, world!\n");

This will cause the memory leak, risk. Now C99 standards have been added to this item, but the authors completed early zprintf () function prototype, haven't seen such standards, also found no such system library functions (2003), and now there are many applications compiler, not support snprintf () function. Due to the reason, in fact we have many people have tried to make the sprintf () function instead. But the author sees fortress version, because of the floating-point conversion feared, if without complete floating-point format conversion functions to remove. This choice is very reasonable, although the work practice in many years, also has been advocated in communication system, and do not use floating-point number shall be equal to an integer operations (for example, by frequency speed or kHz, don't use floating-point 3.5 GHz, power if multiplied by 100 times, without using 0.5 dB these values are convenient, reading, or to convert the memory cannot read specific value directly. But as a system function, without floating-point conversion to remove, this is unacceptable. The authors think that the reason, do is probably due to convert too complex: the conversion function is the system of the library system, function, or source code written by the assembly code, or extremely complex, difficult that didn't have so much energy to study this thing, so the abandoned.

In view of the above reasons, the author in 2003 to develop such a zprintf () function, but didn't get the response, so has been expected cannot put into use. That hasn't seen C99 standards, even C89 standard also haven't read. When compared with the idea is to help, not naive to any

external library functions, and use directly naked C language to write such a zprintf () function, it is ridiculous. And the conversion function is restricted to print the result to convert a caching and can not directly to the default output terminal, therefore also cannot use. But take several definitions and floating-point format conversion function already have.

Today, return overdo to read, find new standard C C99 standards have been occupied % and % z this format identifier (control in 2003, the authors make zprintf () function, they were designed to fetch and user-defined format, C99 standards will they defined hexadecimal floating-point number and length of size_t integer modifier). Thankfully, luckily didn't strongly request will zprintf () function, otherwise the function applied today must was scolded. According to this characteristic, the author revised take several control character @, user-defined format control for Z.

Although this is not conflict, but the authors have some concerns. Just read the new C99 standard authors, those C99 standard, though the drafter than we usually stand high levels of programmers, but it's difficult to say they must not slice errors. For example the length of modified for integral type supplement, KuoZhan three letters, and these type defined, it can actually support to the compiler is not much, BCB6 VC6 (not support, new version of VC and BCB also don't support). So they are actually application is not wide, it is not the decision of Microsoft, it will VC modifier to length of data types, I16 I32 I8 niloti, I64, definition, it doesn't take much to the expansion, the control. For users (here is refers to the user of the compiler to reserve a designer) or some format control, in order to user defined extension, usually for C standard makers is necessary. Take the IP protocol standard makers, they will be reserved for IP address for some big enterprises as private network address, it is very natural thing. In fact I read his today, the compiler to C source code printf () function of standard format conversion control will have different understanding and expansion.

4 Design

liliandi 6/17/05 2:00 PM
Formatted: Bullets and Numbering

Module design ideas, author here is not such a windbag, because the reader to read this patient, generally have certain C programming language ability. How to write the code is not in the book. Here is that there may be many wrote many C code programmers, C standard is how to define the behavior of the C language, is not very clear, also need not clear. Program can compile and run properly, it can be enough. Why is how to formulate C standard tube? But the authors write, because the printf (not) such a kind of function, if not clear how C standard said, then the program code is unable to let people use. So here are necessary C99 a list of the latest standard (C99 standards can http://www.open-std.org/JTC1/SC22/WG14/ website, it is free) for the printf () for the definition of analytical format. My English is not good, no translation. In addition, due to a series of formatting output function, such as printf fprintf (,), snprintf sprintf vprintf, vsnprintf vfprintf, etc,. Although they are different, its core is the same, so the difference in appearance, the author also introduces aspects of no longer list. The significance of these items listed in the convenience of readers, don't lazy online to collect the information, will therefore format conversion points below:

4.1.1 Understand the standard

This program C99 except for the expansion, design thinking is fully compatible with C99

standards. However, due to the time for wchart_t about the relationship to show wide characters format and floating-point representation scheme base for 2, there is no understanding, or haven't fully grasp. But now, basic functions, because do not affect the haven't met a floating-point representation in the machine for the base of 2. Remove the author has realized this defect, the author designs zprintf () format, should be fully support C99 standard formatting output.

For C99 standard undefined (not used to format conversion control, the process of expansion, has the specific content of expansion in subsequent chapters, will be listed. If you find the realization of what C99 does not meet the standard, welcome you to the letter, email address is: tianrzh@hotmail.com, author of the company's email address:tian.ruizhong@zte.com.cn。

4.1.2 VC6 compatibility considerations

In VC environment, the compiler to C below the standard, content of some expansion is not much, mainly including:

F	to modify pointer, used in such pointer 16-bit word processor or longer applications, said far pointer
N	to modify pointer, used in such as one 16-bit word of processors or applications, says nearly pointer
C	character types
S	a string to a wchar_t type
W	to modify a character or string
I64	to modify integers
I32	To modify an integer
I16	integers
I8	char integers

These may be extended for most users can use, but for some users may be used to. The writer is can achieve F, N, and I64 I32, I16, such expansion, I8. In C, S and w type and the modifier of stand-to in-depth study. Notable is, although support % Lx VC6 such format, % Lu, but it will be the understanding with % Lx, for the same meaning, Lu % ireplace namely. For LLX %, % llu this format, VC is equal to interpret % lx, % lu. To display 64-bit integer, must use Microsoft himself modifier: %I64x,%I64u。

4.1.3 VC6 compatibility considerations

L to modify an integer. C99 standard, it can only be used to modify long type double. And it will be expanded into a BCB 64-bit integer type.

BCB6 for C99 standard compatibility, and support the VC format BCB6 (to expand VC6 many available). But for LLX %, % llu still interpret % lx, lu, to show % 64-bit integer, must use % lx. On the other hand, does not support % HHX BCB, % hhu this format, but all will interpret them int type unsigned short data. It is not advisable. So in functional testing,

char prnBytes;

printf("printf bytes: %hhn\n", &prnBytes);

Will lead to memory cross.

4.1.4 glibc compatibility considerations

This upgrade is author glibc development zprintf () function best resource, it provides a complete test cases printf () function is used to verify the basic function of the test, also give the best help. Besides comprehensive support C99 standard (don't support the VC extension), it will also expand for free analytic format selection list what parameters. Selection of control for x$. This is like below:

```
sprintf (buf,
        "%1$*5$d %2$*6$hi %3$*7$lo %4$*8$f %9$*12$e %10$*13$g %11$*14$s",
        i, h, l, d, 8, 5, 14, 14, d, g, s, 14, 3, 14);
```

We see the first output % 1 $5 to $d * * * *, the variable parameters obtained from the list of the first parameter display value, as for $5, variable parameter list from the first five parameters obtained width limits, and then press the decimal value. Although this function is very strong, but the author seems to want to go to see what it is useless, so don't consider the function of plan expansion.

4.1.5 Summarize for Compatibility

Remove the author has been considered not compatible, mentioned above, the compatibility zprintf () function should be able to support. Small changes or can easily. The source in hand, compile. This is zprintf () function and system function of library in different places.

4.2 Taken data design

As noted in the outline, the number of design is designed zprintf () function is the primary motivation is zprintf () function of vitality. Series, When C99 standard has supported snprintf () function of such anecdotal vsnprintf circulate, such as () function is not necessary. And about the function, take several design any compiler are not to be seriously, so zprintf () will still have the meaning of existence.

In summary, the so-called and have taken several design, but with special structure characteristic of C language. In order to avoid programmers in programming are inextricably quoted a structure, listed in each of its members, take several design. When we need to take several rules used to convert the numerical display, as long as we are at a parameter list, and the structure of the address in the format identifier given a conversion formats control. In particular, when a larger project operation, can use C strings of cascade properties define a unified transformation format, facilitate programmers in code, such as when using:

```
#define    IMSI_FMT     “IMSI %1.<4,16@!x”
printf(“the subscriber’s “ IMSI_FMT    “, good luck!\n”, pImsi);
```

Can show a standardized uniformly to address IMSI.

4.2.1 Begin and End for taking data

All take several operators are @ as control characteristics of characters. Use % @ definition, use the fetch % @! Definition of termination take. If the number is used, is directly use % @! Define a fetch behavior.

Take several after beginning, use termination before termination shall take number, use the

default fetch definition fetch.

Such as:

zprintf("%@ %d %d %@!\n", &myStruct, 1, 2);

Is 1 and 2 will be printed, % d % d take several definitions included in the scope, according to take several definitions, execute the myStruct structure analysis.

Have taken several statement, but without conversion formats statement, unsigned integer conversion formats execution. Such as:

short i=-1;

zprintf("%2@!\n", &i);

Result shows: 65535 not - 1, will show - 1, it must be used below the control parameters:

zprintf("%2@!d\n", &i);

4.2.2 Method for taking data

Input parameters is taken () function for the first start address, rather than directly on the pressure of a parameter list take number. Such as the following format is not available:

short i=-1;

zprintf("%2@!d\n", i);

This will lead to visit, please remember that abnormal pointer.

Always take number from a parameter list of addresses, successive to address, high without using skip function under the condition of the number of consecutive, always.

4.2.3 Base format for taking data(Integer and Pointer parameter)

That is, although in C language, the function of the parameters can be a structure, but to change the printf () input parameters, its function of series of parsed into format can participate essence is integral type. Participate in such a character in the stack, it has been promoted to an int type, the final result is pointer is an integer, provide the string string as first introduced into participation. Therefore, this call is not below can be expected results:

```
struct {
int a[2];
}mystruct = {1, 2};

printf("%d %d\n", mystruct);
```

Therefore take several basic thoughts of the so-called, also is only for the contents of the memory, convert to the corresponding integers, finally to after the commissioning procedures or users with different forms of expression. For example, there are signs, unsigned type, hex, 8 into system, pointer, etc. Because of this reason, printf () function is used to represent the integer control is most format, a total of seven (d, I, u, p, o, x x), plus modifier is most length for integral type and set (7) were also there. And there were only two characters of processing, floating-point itself is more complex, so also more, but its essence, is to present four (f, g, e, a case, the rest is mainly). Thus, the C language for the processing is the most attention integers, provide

information is the most abundant). The basic point of departure from several design. Take several format will be parsed first, then it will be an integer as different explanations and conversion.

Here is the key to put, when using the bitwise take several functions, the analytic function always get store the domain, and then the integer associated a bit-field extracted, and finally the conversion. And C99 standard format identifier through similar types of characters (such as d, I, o, u, etc.), fetch statement also by taking several operators "@" end. This expression gives the basic fetch statement formats:

%[Endian for Byte][Bytes to be taken][.[Endian for Bit][Bits to be taken]][Repeat times]@[!]

Client: refers to the given pattern bytes of memory address, its contents are according to the main mode, according to the pattern of small or to analyze. Such as memory address 0x1000 2 bytes starting place, its contents for 0x01:0x02. When the analytical model for main, taken as 0x01 * 256 numerical 0x02 = 258, when + analytical model for small terminal mode, the values will be taken as 0x02 * 256 + 0x01 = 513. If you don't set, and the current operating procedures processor instructions (hereinafter referred to as the system), If set to ">", for the main mode (compulsive network bytes); the sequence, If set to "<", for the small compulsive mode (such as the existing PC system), If "!" With the system, the model.

According to the pattern: bits of fetch bytes integers, start from the top of the corresponding effectively intercept bits or from the least significant bit began to intercept the corresponding number of bits. The former, the latter is called the main mode of small terminal called pattern. JiDuan model reference is the number of bytes take patterns, not system. If the number of bits take, and set patterns are the same number of bytes take mode, Otherwise, if set for ">", for the main mode (compulsive network bytes), if the sequence for "<" and set the compulsive for small pattern. If "!" With the number of bytes, take the opposite pattern.

Take bytes: if not given in the fetch, according to the number of bits are not taken when the declaration, the default integers, bytes, if the number of bits are taken by taking several statement, continue to fetch the bitwise when the number of bytes but not actual implementation bytes, if it does not take the bitwise fetch bytes is taken as the default integer bytes. Otherwise, if take bytes are empty, the number of take the remaining bits and then take a given bytes. Take the value can use bytes in 10 said the string by an asterisk, also can make the "*" from a parameter list. Take the scope of bytes in 1 ~ 8, otherwise will be truncated, namely when more than 8 bytes, will be as 8 processing, when less than 1 byte, will be as one.

Period ". ": take several definitions of bits, if the symbol period, but did not give take bits, taken as 0 (bits with C99 standard precision lead operators).

Take bits: if not given (including leading end also did not appear to take), 8 bytes (1 byte has eight bits), remove all bytes contained bits. Or take a given number of bits. Given the number of bits can be used in 10 said the string by an asterisk, also can make the "*" from a parameter list. Take the bits in the range of 0-64 only less than zero, otherwise, will take its absolute result would

be more than 64 as 64 processing. Note that when the number of bits for 0, 0 not to take, but said it will last bit by bit after all to take several finish take a bit, If the number does not take the operation, it is invalid.

If defines take bits but not defined bytes, take the sequence of bits that take after take bits from the last remaining in the sequence. If you have taken a number of time, use the fetch byte array specifies an integer, synthesized from the required number of bits of integers. If the bits of the remaining bits (or more than 8 times take bytes), take bytes accordingly expanded to accommodate to the bitwise take number of digits to fetch bytes again after implementation.

Repeat the definition see next fetch section.

4.2.4 Repeats and Groups

We know that in C array application is the most common. In order to adapt to the characteristics, use zprintf (), and repeat take will make us work becomes quite simple. As already mentioned IMSI number of samples taken:

#define IMSI_FMT "IMSI %1.<4,16@!x"

zprintf("the subscriber's " IMSI_FMT ", good luck!\n", pImsi);

It is said in the 16 current fetch executive 16 times. Take several repetitions with a comma, "" as a leader. This leads to a problem, for example, when I used to repeat that take the contents of the memory when taking among several forecast.feasibility leave Spaces, so that may need more clearly shows values, such as follows:

01 02 03 04 05 06 07 08

Sometimes, and I don't want take leave space between directly, such as follows:

0102030405060708

How to handle it? Zprintf () function defines repeat take several separator, take several features in repetitive characters ", "behind. Repeat take several separator, can use any character, when your separator and analytical system reserves format character conflicts, please use backslash difinition, such as follows (in C language in the string is two backslash):

zprintf("the subscriber's IMSI %1.<4,\\.16@!x" \n", pImsi);

System in analytic separators, if found a backslash operator, will directly below a character to ignore it, but the special meaning it directly as a separator. Therefore, every character can be taken as the number of repeat separator.

On the other hand, when the number of repeat from addressing, for instance I need to take number 16 times into four groups and so make text more clearly. Therefore zprintf () function and defines the packet unit. Packet unit statement follows the repeat number behind the definition, take again using a comma, "". Besides both means different both in form, exactly the same. Which group of units designated purpose is to take several different with repeated separator.

In not specified number, number of repeated take take several times for default. The number of repeat in designated times without taking group unit designated by default, when the number of data packet unit for infinite (i.e., a group).

Note that in no use packet unit separators, take the number of repeat separator for repeated use several times take minus 1. Packet unit used for the delimiter group unit number minus 1. Only when there is a number of groups or take when unit, will show the number of repeat take

separator or packet unit separator.

Repeat number format. Take：

,[[Delimiter for repeat taking]Times for repeat taking][,[[Delimiter for group]Group nums]]

Repeat take several separator: as stated above, the combination can be arbitrary strings, when it comes to the influence of format characters, like star "*", "@" and the number of operators, comma, "digital" etc, please use the backslash as escape character. But the inclined frame descriptor won't appear on the final show the string.

Repeated several times: take the number that take the total number of repetitive execution definition. The number of repeat fetch values can be used in 10 said the string by an asterisk, also can make the "*" from a parameter list.

Packet unit with repeated separator: take several separator.

Packet unit: the number of repeat take during the execution, switch using packet unit separator rather than repeating take several separators, take the number of repeat inside unit. Packet unit value can be used in 10 said the string by an asterisk, also can make the "*" from a parameter list.

4.2.5 Advanced function for taking data-Skip

In front of a number of basic followed a problem is, when has defined structure, I don't need to display parameters, what should we do? This will involve taking several senior function design: questions (over).

Skip type as the same level with the system type character of function and a YaMen (quite), all of the prefix as defined system ", "#" + "and" - "and size, width and accuracy modifier instructions are no longer function, it only took several Pointers to control or jump jump forward, backward in a given address in the data.

Jump too prior to skip (i.e., fetch address increasing) and reverse jumped back (number) address. Prior to skip that refers to obtain data after operation, the conversion executes here no longer. Below explain jump back on reverse.

Why should have reversed jumped back? The starting point is very simple, as I will show a string. System have default time-released structure:

```
struct tm {
        int tm_sec; // seconds after the minute — [0, 60]
        int tm_min; // minutes after the hour — [0, 59]
        int tm_hour; // hours since midnight — [0, 23]
```

```
        int tm_mday; // day of the month — [1, 31]
        int tm_mon; // months since January — [0, 11]
        int tm_year; // years since 1900
        int tm_wday; // days since Sunday — [0, 6]
        int tm_yday; // days since January 1 — [0, 365]
        int tm_isdst; // Daylight Saving Time flag
};
```

The structure of the fixed format is, but does not accord with what we want show in order of time, such as I want in the year before, when the next month, or as determined I might want to put the number weeks before. However structure definition is not so, according to the order number cannot take the desired goals, or will have to repeat list structure elements. Then reverse jumped back function becomes very effective.

When the reverse jumped back in bytes function take several, functions, namely take several more simple pointer to jump back. Note that, when a number of take after take several pointer offset, has to take the next number to address, if the reverse of take bytes and repeated several times from the previous implementation is used when fetch bytes and take the number of repeat fetch, again fetch and display, the results will be executed with the same fetch. Want to jump more than the execution, must use the fetch bytes and take the number of repeat whiting values greater combination can reach the expected goal. If the address is less than the number of take ginseng when given fetch address, starting as a protective measures, take several pointer will stay in the starting address and cannot perform more to jump.

When using the reverse to jump from the number of functions, the reverse to jump first take bits are greater than judgment has been executed by a number of take the bits, if the conditions are not met, then take several Pointers to jump back from the start, the number of bytes to press to take bits equal bits already reduced to jump bits execution fetch.

If the conditions, the first to judge whether can take bytes for next bits jump. If you can hold, expand to take bytes can hold next jump bits. Then take several Pointers to jump from the bytes, then click to take bits equal to jump from the eight times reduced bytes to jump bits execution fetch.

Note that if the reverse jump back and take the specified number of bits of according to take for 0, and fetch the reverse is similar to jump is not zero, but the number of bits have take bits.

Skip the format of the type of character to lowercase k. When specifying the reverse jumped back when the function, please follow a switch symbols behind k is an exclamation point! "".

4.2.6 Advanced function for taking data-Reverse bit by bit

Although in computer memory, the data is stored. Skim section But in the transmission of data communication, according to the example of transmission bittorrent. We often get confused bits of data transmission sequence. Sometimes need two kinds of modes of sequential contrast.

This involves another function: the reverse.

This function is not dependent on take several operators, namely, not from the given address began, also can be taken to convert data on the reverse of the target output. Therefore the reverse control does not exist to take several format, as the final list of data format, the leading flags control is a backslash '\ \'. With the "+" and "-", "" and" 0 "and" # "similar, regardless of their arrangement, but not repeated use. For example, below the format is feasible:

zprintf("%\\d\n", 65536);

When the integral length for 4 bytes, the results should be 32768. And the following example is not done:

zprintf("%\\\\d\n", 65536);

The output should be \\d.

4.2.7 Compact mode for Bit taking

In C99 standard definition of domain, the space distribution of compact model, whether by using the implementer (namely the compiler manufacturers decided). But in actual use in compact model has rarely. According to the actual implementation code, provides a switch, when using compact model, need to be related to the macro COMPACT_TAKEBITS (open). So, when an integer bit allocation, not enough space domain will retain the remaining bits series, continue to extend to the next integer bit sequences and spatial distribution. Otherwise it will forsake the remaining bits series. This function in reverse jumped back in reverse, when the number of bits already more than bits, compact model will ignore the execution has taken back into bits, and compact model will consider this a factor.

4.2.8 float or double taking

The floating-point number and the basic format take take several format, the difference is similar, because the number of bytes floating-point Numbers in the existing in the prevailing standard is more onefold (special formats of floating-point into consideration, namely only into single) precision floating-point, double-precision floating-point Numbers and long double-precision floating-point Numbers, according to a statement shall take several ignored. Therefore floating-point number format take:

%[Endian for Taking Byte][Bytes for taking][Repeat times]@[!]

Only four byte by byte, 8, and took 10 (or 16 bytes long, rely on the double system defined) bytes. 0 ~ 4 byte by byte 4, 5 ~ 8 byte by byte, August 8 bytes above press long double format.

4.2.9 Characters and string take number

Due to the front of the reasons, and has zprintf () function is not good at wchar_t types of data processing. So for the character data, even in character values between 128-127 ~, conversion program will also will they forced into a char types of data, which brings a little imperfections. But the basic take several definition is still valid.

And the number of different character format is taken when the output string, the program is merely a string pointer. Therefore, to take a string, and the number of bytes statement will take bits are ignored. Take several format string:

%[Repeat times]@[!]

However, there are a number of problems: since taking is the body structure analysis, then there are two kinds of circumstances, is a number of storage space is taken with the function Pointers, string into stack, other parameter is a string, the transformation in the storage space, this is a structure definition.

For a while, before the format and C99 s standard definition, but for the same after a circumstance, please follow a switch in s behind! "" is an exclamation point symbol . The character data from several Pointers directly from the start, rather than for the pointer to fetch address space as a pointer. This exists in the string structures, the distribution of space length, please use a string of output value of precision. It's better to understand, normally, the string is not specified output precision, namely, that does not exceed the length of precision output string. In several mode, zprintf () not allow users to use additional string precision to limit the length of the string trunk parts.

For example, in the structure of the size of the array of characters is 100 bytes, zprintf () cannot accept the conditions for 100 accuracy to output the character string array.

This is mainly due to take several definitions are for the integral type of data, and the value of the largest only to 8 (64). With respect to the current situation, this definition is can fully satisfy requirement of daily.

4.2.10 Recursive for struct

Someone may ask, take several design is produced according to show structure content, so if the structure definition is nested?

The answer is, this problem is considered, but gave up. Because of that, to bring trouble analytic function. After all zprintf () function not compiler, cannot be taken for setting environment by pressure stack. In fact, this application should be much. If you really must show the structure of structure and directly, or print memory do such a cycle analysis. Zprintf () can help you solve streamlined structure element analytical display.

4.3 float(double etc) data conversion

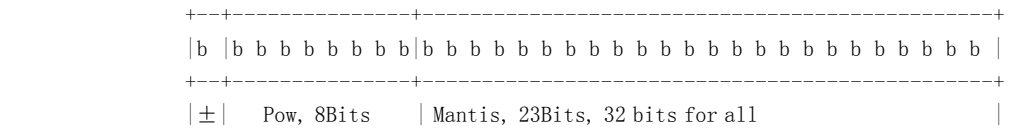
The conversion function is C99 floating-point number, a basic standard function, in all formatting output control format, its use probability accounts for less than 1/3 (because of this, we hold version of the design is simply to ignore it, but this function is the largest and most complicated work, so here on a lecture. The only need to use zprintf () function normal function of the reader will not need to read it.

4.3.1 The floating-point basic format

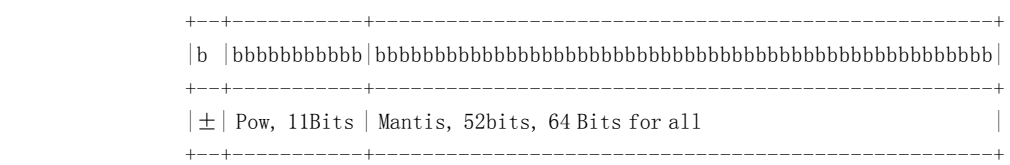
In most popular system into a single floating-point, real, double realnumber and long double real. Long double standard, the real is not with the same type of C int, can choose the compiler or the option. For example, some compiler will grow as a number of double double precision, also may be a 80 double, may also be a 128-bit number of double. In order to simplify treatment, zprintf () will all long double-precision floating-point Numbers for a double conversion are showed. It's hard for some applications, the likelihood is not enough. But so far, the choice is acceptable.

Generally, the floating-point format as follows:

Float:



Double:



For the different modes of CPU, that is the floating-point number one to four bytes or 8 bytes of integers, can effectively from the highest position to the least significant bit intercept the corresponding bits of the corresponding symbols, and the index of terminal.

Normally, because the normalization floating-point Numbers, namely the decimal digits and decimal digits are fixed before. As we use printf () function formats a e the floating-point number, so that the result is only:

1.23e10

while not be:

123e10

Otherwise the conversion function is can't handle, this is the benefits of normalization. The base for 2 floating-point also follow this rule, because it is the most significant bit binary, cannot be 0, therefore, it's the most significant bit 1 is implied. Normally, namely the actual value should be band:

1.xxxxx

There is also an implied, index, namely have offset for real values such as a single 7f, it shows that the practical index 0 is not zero, when the values of 127 when it says - 127. When values for maximum, it says an abnormal floating-point, namely the numerical NaN (to) (a) or terminal to Infinite Infinite) (Inf (0) band.

Special attention should also is, if the index 0 (-) - 127 or, it says 1023 floating-point representation has reached it says precision limit. This is the highest rates of 1 is a hidden effectively. So an integer in memory for 0x00400000 the floating-point and, eventually, the results indicate $0.5*2^{-127}$while not be $1.5*2^{-127}$。

4.3.2 Base method for float data conversion

The number of the basic ideas of the floating-point conversion, finding the floating-point BCB and VC conversion functions, all of the floating-point processor utilization, accordingly its implementation method without good portability. In practical applications, the floating-point coprocessor and can't get the reliable guarantee, different cpus floating-point processor instructions are also different. Because of the embedded system in the industry, so is the design method for such dropped from the start.

The design principle and VxWorks simpler, it will constantly floating-point Numbers multiplied by 10.0, then 10.0 or take the integer part, can be made into various 10. This method is simple, but the problem is not high precision, because the floating-point calculations process, precision loss is unavoidable. In fact, when required in 15 more effectively, Vxworks this practice will not guarantee the reliability of the precision.

Another method can get from glibc, it USES a precision (MPN), to complete the library functions from 2 to 10 base. This method is very complex, really, because the time is limited, the author also didn't fully understand. Moreover, because many precision for itself is complex, said more than a number of bytes to accuracy, said to the mentalistic algorithm is difficult, it is hard to imagine the efficiency of what can be high.

Above two can get and can be roughly read conversion algorithm, need to press for a stack or above 300 bytes of storage space, converting process may appear floating-point string. Because a double value can count the precision scope is 10 ± 307. So according to average floating-point format, write down this space is fully will be enough.

However, the author designs zprintf () function, the function is interrupted, even at also can use. Then don't want it to apply for additional conversion, and hope to pressure stack space as far as possible the space is small. In fact, the author finally achieved this point: for the extra floating-point stack space rigid only about 100 bytes, need not apply for additional conversion space, and precision can guarantee to at least 19 effective number (64 bits, converting 10 into system, i.e. (64 * 3 + 10 = 9) / 20. In this VxWorks and VC below, is not the real thing.

So the author's design idea is what? See the floating-point Numbers:

$$\pm 1.xxxxxxx * 2^{\pm yyy}$$

The difficulty lies in the transformation of the decimal point is, will the one to be dismantled, however, trainer gene monahan 0.5, 0.25 0.0625 such combinations in 10 integers, and then the calculation, the workload unimaginable index. But if we will be wise, the floating-point first convert the normalization of binary floating-point number, make its have scores, but this band:

$$\pm 1xxxxxxx * 2^{\pm yyy - mantisbits}$$

Because of mantisbits digits effective digit calculation, it is easy to translation, can get this expression. Is it will be much easier. This will convert the floating-point Numbers in 10 said, only need to eliminate the index to convert the corresponding parts, with 10 of index, namely the part can do this transformation:

$$\pm 1xxxxxxx * X.XXXX * 10^{\pm YYYYY}$$

Finally get a band for the 10 integers system floating-point number:

$$\pm 1x'x'x'x'x'x' * 10^{\pm YYYYY}$$

In the end, according to the output terminal of the existing system of 10 to 10 digits, effective system for the index of translation. The program is the key index of extinction.

Elimination process is divided into two directions, the index that mantisbits yyy ± is positive or negative -. For timing, will multiply 2 or 4 digit or 8, the corresponding index can decrease correspondingly, conversely as division. However, the band in multiply or division, inevitably accuracy is not. Do? Through the window, make its have fixed translation accuracy, namely when digit times 10 will overflow 64-bit, is divided by 10 and in the terminal eventually converted into the system of floating-point number 10 + 1, the index value of the said that this is the same. In another direction, when rates low threshold than window, will multiply in the final 10 and converting 10 in the index on the floating-point minus 1. Using a remnant of the extra register, and record the loss, can ensure accuracy of conversion of the final result in loss of predictable precision scope, and determine rounding operation (application).

This method is simple and effective, no hard on the PPC floating-point VxWorks efficiency ratio test system function, have twice as hard floating-point PC, efficiency, lower than the system function, only about 1/8 system, but also can be accepted.

4.3.3 Round for float data conversion

This algorithm is realized early the above, haven't read glibc code, also don't know in operation and some extra. Concrete is below:

As to the value in less than 5 digit, all is abandoned, when to rounding bit values greater than 5, will be rounded. And when to rounding is 5 and subsequent value is 0, the rounding a according to whether an odd come to. For even when, abandon, as odd, into a.

This should pay attention to this algorithm, the author spends a great effort to preserve the conversion process lost precision. Otherwise, cannot achieve the above in operation.

4.3.4 float for hex format

And C89 standard, in order to check on the memory conveniently to determine conversion floating-point, whether have problems, C99 standard adds a hexadecimal format floating-point display method. Specifically, it is base band and index, it is 16 C programmers always thinking.

Through testing, authors found a commercial systems support this format. But the authors have realized it.

4.3.5 float data output

By means of the output hexadecimal floating-point its essence, is ZhiShuShi, easy to express. Here no longer. It is the floating-point output. If be % g, the essence that % e format is better index, but if this is the format, % f output string width might have more than 300 bytes

(because the index could be ± 307 power), this is not the precision compensation. If plus a comparison abnormal precision compensation, it is quite difficult to handle the output. Consider the floating-point number, the final output:

xx.xxxxx
0.000……000xxxxx000……00
xxxxx000……0000.0000……0000
x.xxx0……0000e$^{\pm yyy}$

Although this application should be few, but as the system function, we should support designer is demanding application scenarios. This method, through the cache is difficult to realize fixed. Unfortunately have such glibc test cases. In order to realize the function, the author will all floating-point according to characteristics into several parts:

1. Pow
2. Stem(i.e. mantis for float data)
3. Zero pad ahead of dot
4. Zero pad after dot and ahead of stem
5. Zero pad after dot and stem

Note the normalization of floating-point number, also can be used as a not too specifications, such as 123.45 Java floating-point number can be interpreted as 123.45e0, only is 0 (according to this idea is omitted, a floating-point string can easily be converted into double real). Convert into system in October after the floating-point judge the position of the main branch branch, record in what position to fill 0, then to convert cache (pressure) just stack space main stem, in the final part of a string for width and calibration, then fill the unified compensation can be finished 0 width or any of the floating-point output, precision and do not take up too much of a temporary cache or pressure stack space. Because a 64-bit integer, its main parts digit number is 20. Use 28 conversion space will be more (index, add 5 decimal point, namely "0 0" 2 altogether seven).

4.4 User defined format

Zprintf () function of user defined format, just as an example, zprintf () function for the likes of developers to expand its application, because in different industries, the authors believe that can define the format will also has a lot of. This is a software system for early framework, or very helpful, this helps the C code developers will be part of the energy from the debugging liberate the formatting output.

Zprintf () function interface for user-defined format provides parameter parsed functions, namely obtained from several modes of user statement, width and precision definition and the prefix symbols, alignment, extension developers can use well this parsed control format to write the code extension.

User-defined format in uppercase for starting, add a Z extended characters as target model, this subclass user-defined types of most children can achieve 255 characters operators shall not end with (the). Zprintf () function formats to user-defined function of analytical model is the

character of the subclasses, and do not make any conversion. Currently, the defects in the type is not support if the zprintf () existing framework, but not get this feedback.

4.4.1 Frequency

Communication system, frequency is an often used, whether baseband, intermediate frequency variation, or rf, or clock, involve frequency of such a concept. However, at present the relevant frequency that the condition or confusion of industry, the dimensional unity, bring numerous stranded flexible. Like many programmers 800M, GSM frequency may be using 800 such a numerical, but once involve specific frequency, it may be, the use of 0.1 M, there is a floating-point representation precision loss. In the system, and convert many times they could not heal. Examples of this writer's colleagues will ever met, and to solve such fault spent a long time.

Therefore, the author proposed according to the system, after all the physical units using frequency for an integer that kHz, this makes it. But it says 4THz from 1kHz to achieve range, we usually radio waves reach beyond just when 10G transmission. This is quite sufficient said. But the question, such as the speech signal, the GSM bandwidth is what kHz, 9.6 said? Still have such circumstance.

Therefore, based on the floating-point representation method, design of high 4bit as actual frequency index, each rank of gradient is 1000 times, so that the frequency range is quite large. The precision and 28, also can achieve higher than the floating-point Numbers. Normally, we do not use this high-level index, the frequency that is enough. At present, the proposal has been enforced, the supplement to see whether reasonable advice teach.

According to this design thought, frequency as a special type of may, for user-defined format to display. Because the normally, for a rf frequency values, such as frequency, to display 3G 3000000kHz reads is not directly. If it is a 3kHz frequency, will value directly by 1,000,000 May get a 0.000003 GHz display and unfriendly. So using user defined format, it can automatic adaptation.

The authors make the format conversion functions are as follows:

First, YuDeQian redundancy eliminating every three binary decimal digit zero, its value, and by 1,000 dimensional rise.

If the control accuracy requirements of output format for 0, directly output integer.

If no output accuracy, precision for the adaptation: when digit is 10 integer times, convert XXX. D or XXX. Dd format (string shorter are easier to read, conforms to the conventional habit), or by an integer.

If the existing output accuracy requirement, and then the output is the shortest way ontology string requirements of precision. 25MHz as required, 3 accuracy, display the results will be GHz 0.025, rather than 25,000 MHz.

4.4.2 Power

Power is also often used in the communication systems of quantities, generally in decibels number. But the number of many dB, such as dB, dBm dBc, etc. But as the authors advocated a start in general, application of, don't use a floating-point number. Power is a typical example. Normally, communication stations launch power up to 60 ~ 70dBm 1000W (above), and below

the weak signal is 110dBm below are rarely perceive get special instruments (except). Is this a value, if the power value multiplied by 100 times, namely to 0.01 dB for unit, use a short integer is more than sufficient (range, said in 327.68-327.67 dB). - But for ordinary users see that use the dB and not intuitive hundredfold value (actually, we can divide the argument list will show, but after 100 program in everything that is more troublesome), so also can therefore define a user-defined types.

This method not only will then complex, the original, and that by 100 after will value in - 200 ~ 200, beyond the range of between with negative/is infinite.

4.4.3 furthermore

In 2003, the design zprintf () function, such as the main or equivalent shows convenient and IMSI IMEI. But take several design and optimization through full after brewing, such as IP and IMSI format has no necessary to yourself, so cut off part of this code (online have some).

4.5 Extended application

The kernel conversion functions, when we will finish after the printf () function into the application, the author summarizes three classes that can be applied:

1. According to the level of print: this control is the most common type of information, namely when the importance of print stay below the threshold level control, no format and output, saves the CPU efficiency, but also reduce visual pollution;
2. in hot control printout: this print is quite common when debugging, namely a function module, you need to focus on hot part of the content, and display at the end, when will the debugging spam closed. This need through the appropriate design of shield bits;
3. According to the CPU run load decided to control the printout: this is the invention of printing, its purpose is to choose according to business load conversion and output information. Such as setting a second printing, such as information to be observed when the object is activity, can be promptly, and was caught when the object is more frequent or, don't give more effective information. It is like a man asked in education, questioner only care about your general illiterate, asked people don't speak too much to read the book, which was what university diploma, in which read, etc.

In practical application, print is often to points of different control module. So the above three kinds of print, the control can be specified module.

4.6 Programm structure

This section describes the main frame, programming for readers more easily read code content. A total of eight program files, including:

J h, zprintf zprintP. J c, zprintf h j c, QwordLib UserFmtLib j c j c, FloatLib, tst_zprn. C j c, ZPRNSH,

Two headers, zprintf. H is provided to zprintf () function call the function series for users with zprintP statement when, as zprintf j h () function for deep development series contains.

J c program files are zprintf, UserFmtLib j c is a user-defined analytical source format, users can literally replace it. J c is QwordLib zprintf () function in 64-bit integer when used with the library for c j, FloatLib floating-point conversion when used with the library. J c tst_zprn zprintf () function is the function and performance testing code. The basic function of the test section, from glibc source however, thanks to its author. If not, don't know how long does it take to verify zprintf () function to find the usability of related bug. Performance testing and take several tests are completed by the author himself. J c is ZPRNSH zprintf (), whether to use its shell packaging at each developer preferences.

4.6.1 origin of name zprintf

The authors have developed from the formatting output function named zprintf (), the company is due to serve the general called ZTE, ZTE, author Z name field name, the red one loyal, English initials and Z. So in C standard basis, and added a z custom declaration form when the letters, because use z so far, it has not been standard occupied.

4.6.2 Main Content

Generally, the analytical process formatting output function can be divided into three steps (formatting output):

1. Establish control of environment, namely conversion operators such as lead, get that decorate, numerical storage space width and precision, etc;
2. To convert the value obtained;
3. Execute Convertion;

So zprintf () function in formatting output, and three steps: parameter parsed ParseFormat () (), TakeDataByFormat parameters, TakeFloatDataByFormat (), and finally the formatting output OutPutInteger (), OutPutFloat (), OutPutString (), OutPutChar OutPutUserType () (), etc.

Generally speaking, the main thread of program structure is clear. Here is the need that the author once saw early literature in this case the grammar, switch says it's efficiency is not high. So the authors write in the first zprintf () function, parameter parsed to avoid the use case this grammar switch. But in the process of the assembly code that is not so, when there is no judgment, but many of the statements made a directed, so it is how to achieve? After all the documents for OBJECT will find disassembly, the C compiler actually make a search list, according to watch how quickly positioning. So a few years ago, when the case is required, need to use data driven requirement was not necessary. Because no matter how the compiler to do, that is data driving the highest efficiency of insurance.

In the Internet search related material, have a XieYuBo written more deeper understanding of c/c + + and switch statements on the subject placard const with this very detailed description. But

the fact that the said document has biased. When the case is big span between each value, namely the compact formats values, such as 110, 300,2000, switch case will still after multiple judgment. For this reason, this upgrade zprintf () function when the switch, and defines the case of macro data types.

In the format, formatted control is the most complex analysis, improper handling most likely to occupy more CPU utilization. Here the author of all control parameters drew a table zprnCtype []. This will be format compatible, only need to change the form of the content. Code is not changed, because they occupy space is not much.

Compared with glibc program, it USES the thousands of bytes of precision, and several look-up table zprintf () use only about 500 bytes of data, but overall zprintf () method was advanced.

4.6.3 Int64

In 2003, the realization zprintf () function, still not aware of system has provided a 64-bit integer type. So also does not provide the data display 64-bit integer format. Consider now most compilers are provided the function, so zprintf rewrite () function, it directly to the system of 64-bit integer type, the authors write all than 64-bit integer library function to handle more efficient.

Due to C int type selected standard definition of its size is variable, so the processing 64-bit integer and 32-bit integer is a very difficult thing. Not only that, the new C99 standard defines the numerous variable-length data types, such as the pointer, size_t time_t, etc. Although the mainstream of the C compiler all support 64-bit integer, but after all, there is considerable amount of machine, the word length is a 32-bit, even or mainstream. In a 32-bit word processor, although longer compiler provide 64-bit integer, if zprintf () function will work within the word long also into 64bit, due to the final implementation to instruction, or go to the library functions, system zprintf () function of efficiency is difficult. In view of the above factors, zprintf () function in internal processing, still with 32-bit integer for work. Only in particular, will use a 64-bit integer.

Other authors have not clear, at present the mainstream of the compiler, default size, its type integer int is 4 bytes. When the CPU is itself 64-bit word length, will expand to 64-bit accordingly? In addition, int type of data, does not exist only 1 byte?

4.6.4 float

In front of the floating-point transformation has been used more words about the floating-point processing, here no longer. Need account for long, the processing, the number of double zprintf () function are the first to double coercion, then show conversion. Because more than 64-bit integer processing, design additional library. This is miss glibc library, it USES more precision, can handle number more than any of the integer. In fact, in the field of precision, long double use little, because after all, not like the matlab so scientific computing, don't need such a high precision. On the other hand, C standard for long double the number of int type defined as flexible as variable, which give unified handling brings a lot of trouble. Because zprintf () function

of intent is to design in all the C compiler can run. Depends on the specific compiler, the limits of zprintf ().

4.6.5 standard parameter

Careful readers may find that in the source of the package, but every change and list as the formatting output function parameters, such as vfzprintf (), vzprintf () were placed in ZPRNSH. C files, and not in the zprintf file c j. Why is this?

Author be based on such consideration: in the old version of the compiler, also do not support this macro va_copy (). Because zprintf () function will be changed to participate in global structure of the list, in order to reduce the time-varying parameters transfer function call, the pressure stack. So to form the function change itself, zprintf () function interface is very good. But for the user requirements change and pass the list in form of interface compiler does not support va_copy () ragged, may bring error, so will be placed in this kind of function of other documents.

5 Data Description

liliandi 6/17/05 2:00 PM
Formatted: Bullets and Numbering

5.1 Structure Description

5.1.1 Structure for Int64

64-bit integer described the operation environment to determine the structure of the model, so the effectiveness of a macro definition of _BIG_ENDIAN exist requirements. When the compiler support 64-bit integer, timely QWORD will convert the compiler support 64-bit integer type.

```
    typedef struct
    {
#ifdef _BIG_ENDIAN
        DWORD    dwH;                /*  Most Significant Value          */
        DWORD    dwL;                /*  Least Significant Value         */
#else
        DWORD    dwL;                /*  Least Significant Value         */
        DWORD    dwH;                /*  Most Significant Value          */
#endif
    } QWORD;
```

5.1.2 Structure for Convertion

The format control structure, on the first 4 printf () function for shell opening expansion. Behind the entries are for internal use, external may be used. Why not va_list as the parameters of the function transfer (most compilers are so do)? The main reason to consider if the compiler does not support va_copy () the C99 standard macro, compile there could be problems. The author had no further consider the risks, so not using the change.

```
/*----------------------------------------------------------------------------*/
/* Global Control Structure                                                   */
/*----------------------------------------------------------------------------*/
typedef struct tagPrnFmtGlb_T
{
```

```
va_list pVaList;              /*  Variable argument list             */
PCFUNC  pPutCharFun;          /*  Put Char Function                  */
DWORD   dwStreamPara;         /*  Stream parameter handle for output  */
DWORD   dwPutLimit;           /*  Limit bytes for safe output        */

DWORD   dwReadVal;            /*  data taken, value, ptr or LSBs for int64 */
DWORD   dwReadPad;            /*  data taken, value, ptr or MSBs for int64 */
DWORD   dwRemainVal;          /*  Remain bits value for bit taking    */
DWORD   dwRemainPad;          /*  Remain bits value for bit taking    */

BYTE    bType;                /*  Destination data type: d, f, s, ... */
BYTE    bSubType;             /*  Subtype for convertion             */
BYTE    bModi;                /*  modifier(char, short, long,...)     */
BYTE    bUserTake;            /*  user taken declared?               */

LONG    lWidth;               /*  field width                        */
LONG    lPrecision;           /*  precision                          */

BYTE    bPreChar;             /*  Leading char, ' ' or '0' or none    */
BYTE    bLeftPut;             /*  Left alighment?                    */
BYTE    bSign;                /*  Show Sign?                         */
BYTE    bAlter;               /*  Alternate show Prompt string flag?  */

BYTE    bTakeBytes;           /*  Bytes once taking                  */
BYTE    bTakeBits;            /*  Bits once taking                   */
BYTE    bTakeAlt;             /*  Alternative taking for string       */
BYTE    bRemainBits;          /*  Available bits for bit taking       */

BYTE    bSysLittle;           /*  System endian mode as little?       */
BYTE    bBytLittle;           /*  Byte taking endian mode as little?  */
BYTE    bBitLittle;           /*  Bit taking endian mode as little?   */
BYTE    bBitMirror;           /*  Bits mirror?                       */

BYTE    *pbTakeStart;         /*  start address for taking data       */
BYTE    *pbTakeAddr;          /*  current address for taking data     */
LONG    lTakeTimes;           /*  taking times                       */
LONG    lGroupUnit;           /*  taking times for each group         */

BYTE    *pbMultDelim;         /*  delimiters for multi taking         */
LONG    lMultDLen;            /*  delimiters length for multi taking  */
BYTE    *pbGroupDelim;        /*  delimiters between taking groups    */
LONG    lGroupDLen;           /*  delimiters length for group taking  */

BYTE    baTempBuf[MaxStemLen_M];/*  temp buffer for string take        */
```

```
} PrnFmtGlb_T;
```

5.1.3 Description for float data

Floating-point number 16 byte occupying describe altogether. Content as follows:

```
#define     FltRoundNone_M          0           /*  no round check begin    */
#define     FltRoundUndr_M          1           /*  round up under-level    */
#define     FltRoundCrit_M          2           /*  critical for round up   */
#define     FltRoundOver_M          3           /*  round up completed      */

/*-------------------------------------------------------------------------*/
/* structured floating-point number, Val = (+/-)qwMantis * bBase ^ lExp     */
/* bNegtive shows negative sign, bMantisDigs means significant digits, it's */
/* helpful to calculate valid digits number                                */
/*      (Private Float) Tenzsure Tian@2009-3-7 9:05:08                     */
/*-------------------------------------------------------------------------*/
typedef struct
{
    BYTE    bNegtive;               /*  negative value flag                */
    BYTE    bMantisDigs;            /*  mantissa digits                    */
    BYTE    bBase;                  /*  base, 2, 10, 16, etc.              */
    BYTE    bRound;                 /*  round up flag                      */
    LONG    lExp;                   /*  exponent val                       */
    QWORD   qwMantis;               /*  mantissa                           */
} PFLOAT;
```

5.2 Global Vars

liliandi 6/17/05 2:00 PM
Formatted: Bullets and Numbering

All variables are static can only read not to write. Includes the following:

static BYTE baLowDigs[20] = _T("0123456789abcdef");

static BYTE baHihDigs[20] = _T("0123456789ABCDEF");

These two variables used to convert hex value, converting speed (don't judge values between 0-9 in 10 ~ 15 or in between.

static BYTE baNullPtr[] = _T("(nil)");

static BYTE *cSignTable[] = {NULL, _T(" "), _T("+"), _T("-")};

It's a global variable used as an integer value or pointer conversion. Static variables within the function.

static BYTE *baErrFlt[4] = {_T("inf"), _T("nan"), _T("INF"), _T("NAN")};

Floating-point conversion.

static BYTE zprnCtype[96][3] = {…};

Format type look-up table, convenient fast positioning.

const static QWORD qwDecDigitsBorder[] = {…};

The decimal boundary. The integer array used to determine the final using decimal digit figures for said. For only one digit 1 ~ 9, 10-99 need two digit, 100 ~ 999 need three digit.

6 Self-Test And Analysis

liliandi 6/17/05 2:00 PM
Formatted: Bullets and Numbering

Due to the limited time and resources, all tests under only based on vc + + 6.0 development platform, BCB6.0 development platform and VxWorks5.5 development platform. One VxWorks5.5 embedded system based on the test and only MPC852 processor chips (with floating-point processor).

After running on the platform in different zprintf () function gets the following series.

6.1 Function Test

Functional test, the test cases is given by the test function: glibc TST printf -. Using the results are expected to develop code system and the printed results were compared. In the current format extension oneself glibc outside, the rest of the function has been fully meet the requirements (precision and accuracy are ensured), compared with other systems function test result, it is not so ideal. As VxWorks and VC floating-point precision, the length of the modifier is often not support, etc.

Can say, the author has mentioned in the wchar_t character type 2 (processing and floating-point base is not within the machine that 2), the realization of the function of the program, the program is the best (see than not, because the test cases glibc from there, but not verify its own).

6.2 Performance Test

Performance testing just use floating-point and integer print format to compare each system under different ways of realizing printf (series) function performs format conversion takes time.

In order to ascend zprintf () function, the author endure to the performance of several sleepless nights. Early tests, the PC operating environment, VC system and not the same BCB system, the system performance comparison test cases. Then after this situation, is improved.

First, the performance test results zprintf () function transformation performance frustrating. Because the system of floating-point conversion efficiency is zprintf () function of 300 times. What is this idea? This is your trip depends on two legs walk (5km/h), usually sit in the family, but has been faster than supersonic plane of transportation (1200km/h). So the author spent so hard to develop, not worthless things actually frustrating? Swallow not to descend this tone, through contrast, suddenly discovered a problem: the integer conversion system function with floating-point conversion, the difference is not big, the efficiency of only three times. Imagine, if the algorithm, regardless of how you are advanced, nor may within such a short period of time in a binary floating-point converted to a decimal floating-point number. Read the back of the assembly code BCB, it can be found in the process of transformation system function, the floating-point instruction, such as floating-point and floating-point except etc. And this is the beginning of in the

design of the author, because not consider zprintf design () function, is to have the floating-point machine in embedded system print floating-point, all of the floating-point and floating-point operation is not only soft hard floating-point instructions.

Not only that, when the source program VC compiler to test this code, and use speed optimization option, the exciting: the integer conversion efficiency even higher than the system function, less than half the time transformation system. And floating-point conversion although slower than the system, but also only about four times. And this result in BCB system with the operation result, below the compiler BCB apparently slowed the user program execution speed. Therefore this point of view, the compiler than VC BCB compiler works. Support the GCC BCB (standard is the latest)

Then the authors refer below in BCB, floating-point conversion efficiency low to the incredible BCB compiler, found the very stupid thing: namely BCB compiler in dealing with 64-bit integer division, are also promoted the dividend will be a 64-bit integer, executive 64-bit integer division. Want to know, in 32-bit cpus processor, there is no 64-bit integer division (addition, subtraction multiplication are not programmers write oneself needs) related instructions or code to complete these basic operations. Actually the floating-point, because it shows in 2003, write the code. Addition and multiplication, it is easy to realize subtraction, just a few instructions. And besides, it need not by rule by bit by bit. Actually perform a full 64-bit division, to walk more than 300 64 circulation, instruction. The assembly code below BCB:

```
_lludiv()
+00h    0x004083ab : push ebp
+       0x004083ac : push ebx
+       0x004083ad : push esi
+       0x004083ae : push edi
+       0x004083af : mov ebx,[esp+0x14]
+       0x004083b3 : mov ecx,[esp+0x18]
+       0x004083b7 : or ecx,ecx
+       0x004083b9 : jnz _lludiv+0x18
+10h    0x004083bb : or edx,edx
+       0x004083bd : jz 0x4083ee
+       0x004083bf : or ebx,ebx
+       0x004083c1 : jz 0x4083ee
+18h    0x004083c3 : mov ebp,ecx
+       0x004083c5 : mov ecx,0x00000040
+       0x004083ca : xor edi,edi
+21h    0x004083cc : xor esi,esi
+23h    0x004083ce : shl eax,1
+       0x004083d0 : rcl edx,1
+       0x004083d2 : rcl esi,1
+       0x004083d4 : rcl edi,1
+       0x004083d6 : cmp edi,ebp
+       0x004083d8 : jb 0x4083e5
+       0x004083da : jnbe 0x4083e0
+31h    0x004083dc : cmp esi,ebx
```

```
+          0x004083de : jb 0x4083e5
+          0x004083e0 : sub esi,ebx
+          0x004083e2 : sbb edi,ebp
+          0x004083e4 : inc eax
+          0x004083e5 : loop _lludiv() + 0x23
+          0x004083e7 : pop edi
+          0x004083e8 : pop esi
+          0x004083e9 : pop ebx
+          0x004083ea : pop ebp
+30h       0x004083eb : ret 0x0008
+          0x004083ee : div ebx
+          0x004083f0 : xor edx,edx
+          0x004083f2 : jmp 0x4083e7
```

From this string assembly code, we can see that, when the divisor is a 32 digits (0x004083b7: ecx value is 0), without any special treatment. It just judge when the dividend and the divisor is a 32-bit integer only execute simple division instructions, and ignore when the divisor is a 32-bit integer, also have shortcut, need not go that more than 300 the theory instruction. This is caused by the conversion function in BCB environment of floating-point conversion efficiency low to the root cause of the incredible! In fact, when the use of 64-bit computing system of instruction, and use directly oneself write 64-bit library functions, add, subtract, multiply and divide, efficiency will have a few times.

Therefore, the authors try to use compiled by a 64-bit integer 32-bit integer division operation function as follows:

```
QWORD QwordDividNum(QWORD qwVal, BYTE bVal, BYTE *bResidue)
{
    QWORD     qwRet;

    __asm
    {
        xor ecx,ecx
        mov cl,[ebp+0x10]
        mov eax,[ebp+0xc]
        xor edx,edx
        div ecx
        mov ebx,eax
        mov eax,[ebp+0x8]
        div ecx
        mov [ebp - 8],eax
        mov [ebp - 4],ebx
        cmp dword ptr [ebp + 0x14], 0x00
        jz   DividNumRet
        mov eax, [ebp + 0x14]
        mov byte ptr [eax],dl
    }
```

```
DividNumRet:

    return qwRet;
}
```

There is a limit the divisor bytes (because for floating-point decimal conversion, only by 10) multiplied by 10 or, in fact, the assembly code applies the divisor is a 32-bit integer also can completely. PC instruction is executed instructions, integer division, after completion, the remainder is in registers. And the use of PPC instruction system is not so good. Although the C language realization 64-bit integer divided by a number of bytes of the algorithm, and also can be expanded into the divisor is a WORD (2 bytes), but cannot be extended into the divisor is a 32-bit integer.

This 64-bit integer division, after the problem of low efficiency in VC compiler and VxWorks compilers validation doesn't exist. Finally author in three kinds of environment verification system transfer function and zprintf () function gets the efficiency, the performance under the table.

	VCsys	VCzprn	BCBsys	BCBzprn	VxSys	VxZprn	VxOss
%.24g	4.23	15.72	2.25	14.17	30.86	14.6	X
%a	X	2.54	X	3.38	X	4.34	X
%lx	1.53	0.90	1.13	0.78	0.60	1.03	0.76
%lo	5.03	1.00	1.31	0.86	0.62	1.08	0.77
%ld	4.91	1.10	1.15	1.06	1.32	1.08	0.80
%lu	1.76	1.06	2.00	1.64	1.38	1.07	0.80
%llx	7.94	0.99	9.09	0.99	0.69	1.19	X
%llo	5.38	1.03	19.8	1.01	0.73	1.32	X
%lld	6.06	2.54	13.06	3.46	2.81	3.65	X
%llu	6.37	2.59	13.15	3.52	2.81	3.58	X

Data in table 1.2345678e-300 refers to such a floating-point number, the executive 1,000 times the left column of the table takes time to convert the milliseconds (X system does not support this format), is the actual execution 1 second test execution of the total number of computation can be obtained the corresponding conversion, finally milliseconds. Due to the CPU does not fully occupied by testing procedures, there exist some interference between, such as timing task, etc. But can see a circumstance, because can verify, basically is in the order of magnitude. Also note the author has no support system transformation format and its equivalent format, for example, VC, but does not support LLX % % I64x BCB, only not only supports and support LLX % I64x %, the author or % Lx is equivalent to them as LLX % of the table, and then in comparison.

From this list can be seen in VC environment, besides floating-point and integer conversion efficiency than system widely. While in BCB environment, because the mistake, it BCB itself in 64-bit integer display, performance is far less than zprintf () function. But there is a BCB strengths, that is the floating-point BCB surprisingly high efficiency, even with the 32-bit integer, it can display in the near a magnitude (than), lu % of its own 64-bit integer showed high efficiency, it is very strange. Author to now don't understand how such skill BCB can make the floating-point conversion efficiency is so high. Must be in BCB changed this 64-bit division of defects, performance will be improved: despite 64-bit division, its efficiency than defect, and floating

precision VC than VC.

In the test, VxWorks environment due to the CPU is very low, presenting itself, but only 50MHz instruction system is different, so the PC and the time is calculated by 40 times, because the PC is 3GHz investigations may not 100% efficiency. This can be a nearly equivalent value: in this table, all zprintf () function is similar to the efficiency of volatile, and can be used as a reference. The system function, VxWorks compared the floating-point conversion efficiency zprintf () function only half the unreliability and precision. In fact, if you vxWorks system, print a precision for 20 more floating-point, system can give you a series of end is not zero the display, but that is nonsense. Remove floating-point, vxworks system function in dealing with 8 in display and hexadecimal displayed with special skill, seems fairly high efficiency of it, close to zprintf is twice the function. But in 32-bit integer, and as the author zprintf () function. For 64-bit integer zprintf () function, lower than the system, but can accept.

Besides using the system function as reference, the author also performance introduced a stronghold printf () function of edition, VxOss is a performance of the data. This version of the printf () function, performance zprintf () function than 1.5 times 1.3 to high, it is easy to understand: because this function has no function division into son function, and went on PPC stack cost function is more. Although it's performance is better, but due to the weak, so the use function of the meaning is not big, listed here are for reference only. The program source code in the bag, provided it is my colleague company, don't know the reader's praise is not provided, so the author's name, the author hopes this saw me.

Based on the analysis of the writing, the zprintf () function conversion efficiency, generally close system function even than the system function often conversion efficiency, especially the invention of a floating-point coprocessor without converting to floating-point string, is the most advanced (born in 2003)

7 Program Package and Copyright

7.1 Program package

7.2 Copyright

The authors designed zprint formatting output conversion functions, occupy part time work, also have a large part of the work is completed by do unpaid overtime time, so the author ministering zte entitled to free to use it, at the same time, you can also use free. Finish this work, purely personal self motivation. But you shall not subject to the consent of the circumstances, it cannot do without modification, the code is fake your own work. If you find application in it have any bugs and Suggestions, welcome you the E-mail is discussed in this paper. The email address is: tianrzh@hotmail.com, email address: tian.ruizhong@zte.com.cn company work.

8 Referenced Source Code

8.1 VC Source Code

The conversion function actually VC kernel output is the output files (j c), is located in VC directory Microsoft Visual Studio VC98 \ SRC \ \ CRT below

8.2 BCB Source Code

The kernel function is BCB conversion __vprintert (), eventually CBuilder6 \ returned documents in IO \ \ \ Rtl returned vprinter \ c j. Floating conversion functions: __realcvtt is a macro, actual effect should be __realcvt or __realcvtw series function. But did not find the C source code, can use _gcvt () function performs floating-point conversion, this application form but is like below, g, % :

_gcvt (& dfVal, 24, buf),

That provides a double number address, provide precision, provide conversion cache. This interface doesn't work well, like # # function cannot play a role. Vprinter. C source code below:

8.3 Linux Source Code

In the file c j libs \ vsprintf (vsnprintf () function). Don't support floating-point operation system itself, because the printf () function is not needed. I understand that it is an operating system, not to develop applications, so don't provide users with the interface. source:

8.4 glibc Source Code

The core code file exists stdio - common VFPRINTF \ C (j), floating-point conversion in printf_fp. C source file. Floating conversion use hard, but the use of floating-point number to compare the accuracy of library, gradually approximation to get the final result. High precision can reach the conversion, for example, but that of the conversion, and the algorithm is not complex, the authors consider to abandon.